Novel Simulation Concepts for Buildings and Community Energy Systems based on the Functional Mock-up Interface Specification

Edmund Widl*, Benoit Delinchant†, Svea Kübler‡, Dan Li**, Wolfgang Müller*, Victor Norrefeldt‡,
Thierry Stephane Nouidui§, Sebastian Stratbücker‡, Michael Wetter§, Frédéric Wurtz†, Wangda Zuo**

* AIT Austrian Institute of Technology, Vienna, Austria
† G2ELab – Grenoble Electrical Engineering Laboratory, Saint-Martin-d'Hères, France
‡ Fraunhofer-Institute for Building Physics IBP, Holzkirchen Branch, Valley, Germany
** University of Miami, Coral Gables, FL, USA
§ Lawrence Berkeley National Laboratory, Berkeley, CA, USA

***Abstract*—Research and development in the fields of building technologies and community energy systems have caused in recent years a transition away from stand-alone components towards dynamically interacting systems. However, the precise modeling and simulation of such complex cyber-physical systems proves challenging for the established simulation tools. This paper illustrates the applicability of co-simulation and model exchange concepts based on the Functional Mock-up Interface (FMI) specification for the simulation of buildings and community energy systems by comparing several state-of-the-art approaches. The presented applications thereby demonstrate the suitability and relevance of such modular and flexible simulation concepts for these fields. At the same time the importance of a common simulation interface is emphasized, which allows the reuse of individual components for a diverse range of applications.**

***Index Terms*—buildings simulation, community energy system simulation, co-simulation, model exchange, Functional Mock-up Interface, FMI**

I. INTRODUCTION

Buildings increasingly comprise cyber-physical aspects due to the technical requirements imposed by the need to reduce energy and peak power while maintaining a high level of occupant comfort. Even though there is a wide variety of building simulation tools available [1], it still remains challenging to incorporate these novel features properly into a model-based design approach. The modeling of buildings and community energy systems (e.g., district heat networks or smart grids) requires the integration of multiple domains (e.g., air-flow, thermodynamics, controls, indoor environmental quality or electrical grid), which are not all covered sufficiently by any of the established tools.

The *IEA EBC Annex 60*[1] collaboration provides an international platform for partners from academia and industry to overcome these limitations. It aims at the development of a new generation of computational tools for building and community energy systems by employing state-of-the-art modeling and simulation concepts. It focuses particularly on the utilization of the *Modelica* [2] language and co-simulation and model exchange approaches based on the *Functional Mock-up Interface* (FMI) specification [3].

Within the scope of the IEA EBC Annex 60, co-simulation approaches are being developed and investigated in order to target problems that have to be addressed at the system-level or require the vertical integration of domain-specific simulation tools. For instance, performance optimization studies for community energy systems need a system-level view that integrates domain-specific tools beyond the scope of typical building simulations. Another example is the integration of tools for computational fluid dynamics or daylighting, for which equation-based models may not exist or may not be suited.

For interfacing the individual components, the FMI specification has been selected, which defines a standardized API and model description for both co-simulation and model exchange. FMI has been selected as it is a non-proprietary, industrial-strength specification, developed by both academia and industry, that will facilitate the technology and knowledge transfer between the building performance simulation community and other communities, such as from controls, power plant or electrical system modeling.

II. SURVEY OF SELECTED SIMULATION APPROACHES

This section presents several R&D activities associated with the IEA EBC Annex 60, which focus on the simulation of buildings and community energy systems. Within this special context, they examine the feasibility of co-simulation and/or model exchange based on the FMI specification.

A. Integration of zonal airflow models

An example application of dynamic co-simulation has been implemented at the *Fraunhofer-Institute for Building Physics*

---

[1] See http://www.iea-annex60.org/.

978-1-4799-4702-7/14/$31.00

IBP, where the building simulations tool TRNSYS [4] and the Modelica-based zonal model for airflow computation *VElocity Propagating ZOnal model* (VEPZO) [5], [6] have been coupled. The underlying models allow for instance to compute the transient prediction of the airflow in an atrium, which is indispensable for analyzing the influence of an atrium in terms of building energy performance. In this case, only the airflow inside the atrium is modeled in detail while the thermal behavior of the rest of the building is approximated by the computationally in-expensive multi-zonal building models provided by TRNSYS.

A generic FMU import interface for co-simulation has been implemented [7] for TRNSYS with the help of a new simulation component—referred to as type—that allows the integration of any FMU into TRNSYS, see Fig. 1. With the help of this type also VEPZO models can be imported as FMUs, see Fig. 2. At this interface TRNSYS communicates wall temperatures to VEPZO. From this information VEPZO computes the resulting airflow and passes the wall heat flux back to TRNSYS.

Exporting VEPZO models as FMUs for co-simulation also revealed limitations of current tool support. For instance, VEPZO models exported as FMUs for co-simulation with the help of Dymola 2013 failed to reproduce the same results as VEPZO models simulated directly in Dymola 2013. Clearly, the exported solver was not able to cope with the task at hand. This demonstrates the demanding requirements on FMUs in the field of buildings simulations due to the complexity of the used models. In order to allow for comprehensive co-simulation studies the tools used for creating FMUs have to be accordingly mature, advanced and flexible, e.g., providing model-optimal numerical solvers.

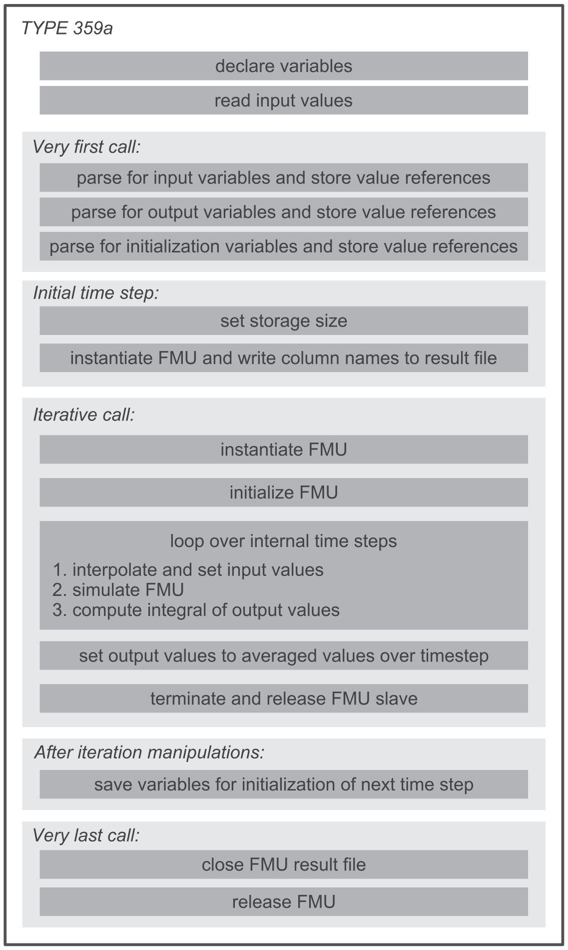

Fig. 1. Flow chart of the new TRNSYS type, implementing the FMU for co-simulation import.

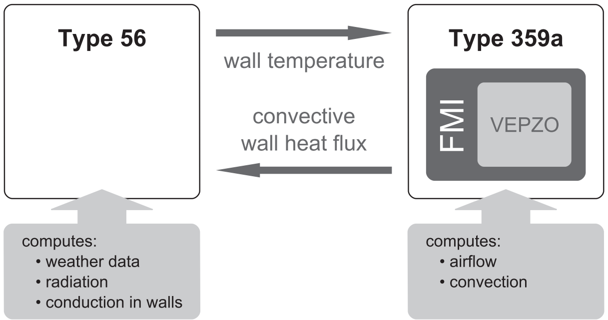

Fig. 2. Schematic integration of a VEPZO model into TRNSYS.

B. Integration of computational fluid dynamics simulations

Energy efficient ventilation design, such as mixed-mode ventilation and displacement ventilation, can lead to stratified air distributions inside buildings. The available multi-zonal airflow models in current Modelica-based building simulation tools do not consider stratified airflows, since all of them rely on the *well-mixed assumption*. However, by numerically solving the Navier-Stokes equations and other governing equations on fine grids, *Computational Fluid Dynamics* (CFD) tools can calculate air velocity, temperature distributions or contaminant transportation in detail. Meanwhile, Modelica-based building simulation tools can provide information concerning walls, windows or supply air diffusers that are critical boundary conditions for CFD calculations. Thus, the co-simulation of Modelica-based tools and CFD tools is a feasible approach to study the design and control of buildings with stratified air distributions.

Studies conducted at the *University of Miami* have already in the past focused on the co-simulation of established building energy simulation tools and CFD [8]. However, in contrast to these previous studies, the major challenge addressed in an ongoing project is the development of interfaces that enable the data transfer between the acausal input/output of Modelica and the causal input/output of CFD tools. Within this context, the feasibility of co-simulation export APIs for CFD applications based on the FMI specification is currently investigated, as they would allow the direct integration of CFD models into most of the available Modelica simulation environments.

Fig. 3 shows a Modelica model that facilitates proof-of-concept studies for this co-simulation approach. The open-source Modelica *Buildings* library [9] was used to create a room model that also integrates an advanced indoor airflow

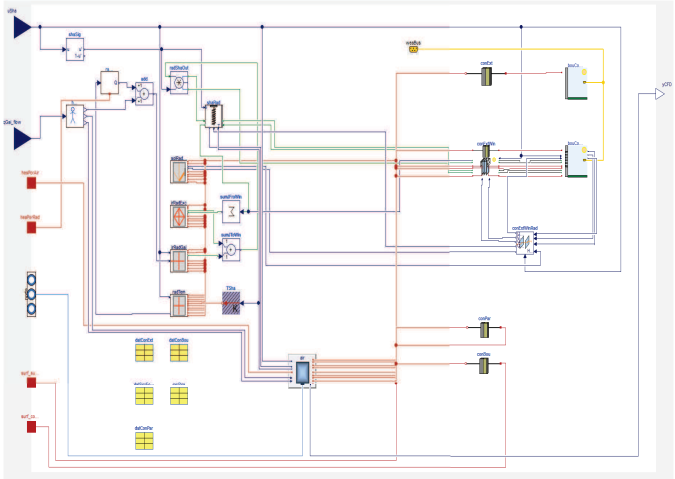

Fig. 3. Diagram of *Buildings.Rooms.CFD* model.

model via coupling to a CFD application. For the indoor airflow simulation, the *Fast Fluid Dynamics* (FFD) [10] simulation tool was selected because it applies different numerical algorithms that can solve the same governing equations typically 50 times faster than other established CFD tools. This speed-up can be even increased further by an additional factor 30 by running the FFD simulator on a graphics processing unit, resulting in a total speed-up of 1500 [11].

Fig. 4 shows the result of such a case study on the natural convection in an empty room. The flow is driven by the buoyancy force caused by the temperature difference on the opposing east and west walls. The heat transfer through the wall was calculated with the help of Modelica, whereas the airflow was calculated with the help of FFD, using a $10 \times 10 \times 10$ spatial grid and a time step size of 0.1 s. The data was synchronized every 60 s and the setup showed a performance 20 times faster than the real time on a laptop computer.

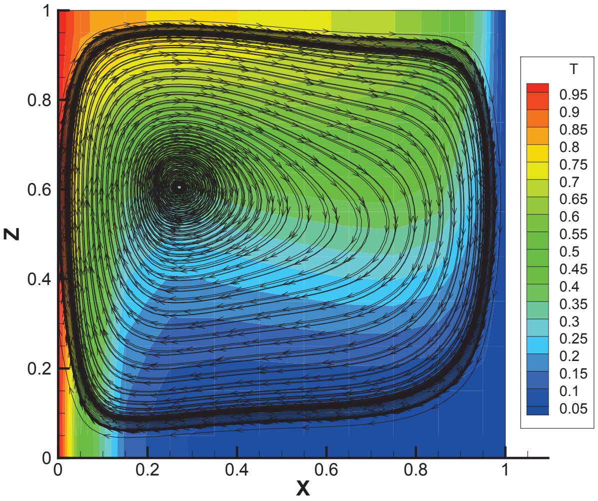

Fig. 4. Temperature contour and streamlines results from a FFD simulation of a natural convective flow in an empty room.

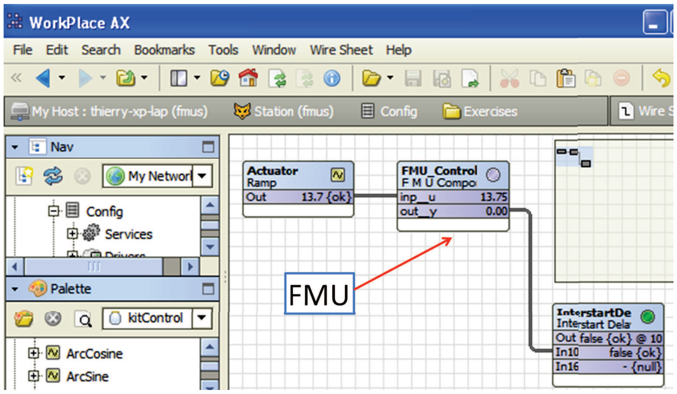

Fig. 5. FMU import interface for co-simulation in Niagara^{AX}.

C. Integration of building management systems

An FMU import interface for co-simulation has been implemented for the building management system Niagara^{AX} [12] at the *Lawrence Berkeley National Laboratory* (LBNL). The intention was to provide a building management system with a robust pathway for linking measurements and simulations to support building operation. Niagara^{AX} is a Java-based framework for controlling and managing diverse devices across a building in real-time. It is based on the open-source *Building Automation Java Architecture*, which provides a vendor neutral, internet-enabled, object-based framework for building automation systems. The FMU import interface that has been added to Niagara^{AX} allows the import of simulation models or simulation tools exported as FMUs. These FMUs appear then in the framework as input/output blocks, which can be connected to other blocks (see Fig. 5). The FMU import interface allows to create a simulation model during the design of a building, export the model as an FMU for co-simulation, import it to Niagara^{AX} and then finally link the model input to measured data. The design model can then be used to compute expected energy consumption, which in turn can be used to compare measured with expected performance. For closed loop control, model outputs can be connected to actuators.

D. FMI support for EnergyPlus

EnergyPlus [13] is a whole building simulation tool designed for annual performance analysis of buildings, which is widely used in the building community. It was neither intended to be used for the simulation of dynamic responses of HVAC and their control systems, nor for detailed airflow modeling or daylight simulation. To overcome this limitation, an FMU import interface for co-simulation for EnergyPlus has been developed at the LBNL. The interface has been successfully utilized to couple a room model implemented in EnergyPlus with an HVAC system implemented in Modelica and exported as an FMU [14].

Although interfaces and simulation programs exist that facilitate the coupling of EnergyPlus with other simulation tools, they still require users to be familiar with EnergyPlus. This is sometimes not desired, for example when developing a

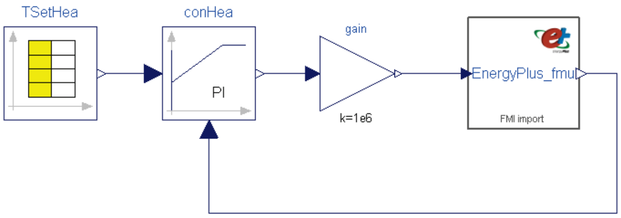

Fig. 6. Linking an EnergyPlus model exported as an FMU with a PI-controller implemented in Modelica.

controller or an HVAC system. In such a case a user may want to take advantage of the visual editor and plotting capabilities of a Modelica modeling and simulation environment, while using an input/output block for a building model that takes as input the control action and outputs a sensor signal. For this reason, the *EnergyPlusToFMU* program [15] has been developed at LBNL, which facilitates the export of EnergyPlus as an FMU for co-simulation. It takes as input an EnergyPlus input file, an EnergyPlus data dictionary and an optional weather file and then creates an FMU for co-simulation. This FMU can be imported in a simulation environment such as Dymola, where it will appear as an input/output block which can be used along with other models. Fig. 6 shows a model which contains such an EnergyPlus FMU. This FMU is connected in Dymola to a Modelica model of a PI-controller. The FMU sends the actual room air temperature to its output which is used by the PI-controller to compute the heating or cooling energy required to maintain a certain set-point.

E. FMI support in the Building Controls Virtual Test Bed

The Building Controls Virtual Test Bed (BCVTB) is an open-source middleware developed at LBNL, which is designed to support co-simulation of different simulation tools [16]. The BCVTB can also be used to link simulation tools to hardware through its BACnet and Analog/Digital interfaces [17]. It is a special configuration of Ptolemy II [18], with the addition of actors and examples relevant for the buildings community. A recent extension of the BCVTB is the integration of an FMU import interface for co-simulation. This interface allows users to import and couple simulation tools which do not support BCVTB's own simulation API, but can be exported as FMUs. It allows the BCVTB to be used as a master algorithm for co-simulation using the *Synchronous Data Flow* domain [18]. It provides a pathway for FMUs to be linked to hardware, and a graphical user interface for linking and simulating FMUs.

F. A co-simulation framework based on the FMI++ library and Ptolemy II

A versatile co-simulation environment relying on the FMI specification for interfacing individual simulation components is currently being developed at the *AIT Austrian Institute of Technology*. The core functionalities for the direct handling of FMUs are provided by the FMI++ library [19], whereas Ptolemy II is utilized to coordinate the execution and data exchange between the FMUs. See Fig. 7 for a schematic overview of the interplay between the two.

The FMI++ library is a collection of generic utilities that facilitate the handling of FMUs within a simulation framework. On the one hand it offers solutions related to the low-level access of FMUs for both model exchange and co-simulation, such as model description retrieval, dynamic model initialization or data access using variable names. On the other hand it implements several high-level functionalities, like advanced event handling, event prediction or numerical integration. With the help of these functionalities, FMUs for model exchange can for instance be used as independent components within a simulation framework, completely equipped with a solver and able to predict internal events.

The FMI++ library also provides a flexible and generic mechanism to implement FMI for co-simulation export interfaces for simulation tools. This feature especially targets simulators that do not provide their own simulation API, but allow to include user-defined types. The feasibility of this method has has been successfully demonstrated by implementing an FMU export interface for co-simulation for TRNSYS [20].

Within the co-simulation environment, Ptolemy II is used to coordinate the execution of the FMUs and the data flow between them. With the help of the FMI++ library, FMUs for both co-simulation and model exchange can be included as independent simulation components. This approach uses Ptolemy II's *Discrete Event* domain [18] as model of computation for handling the execution and data flow, which allows a more flexible handling of events, since simulations are not restricted to fixed time step semantics [21]. However, strictly speaking Ptolemy II does not provide itself the co-simulation master algorithm, since several tasks that are typically handled centrally by a master algorithm are in this approach implemented in the underlying FMI++ layer.

Fig. 8 illustrates an example application for this approach. Shown is Ptolemy II's graphical representation of a model combining a building and a district heating network. The build-

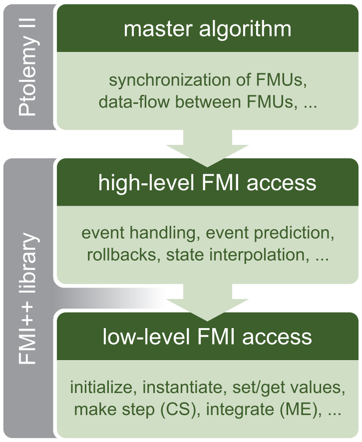

Fig. 7. Tools hierarchy of Ptolemy II and FMI++.

ing was modeled with the help of TRNSYS and exported as an FMU for co-simulation (using the above mentioned FMI++ functionalities). The district heating network was modeled in Modelica and exported as an FMU for model exchange. This demonstrates the seamless integration of both types of FMUs—model exchange and co-simulation—within the same model.

G. Modeling and simulation interoperability issues beyond the scope of the FMI specification

Even though the FMI specification is dedicated to solve interoperability issues, there are several aspects where it falls short of a solution, e.g., semantic interoperability or support for models based on Differential Algebraic Equations (DAE). In approach led by the *Grenoble Electrical Engineering Laboratory* (G2ELab) a software component solution has been proposed that aims at coping with all these interoperability issues.

1) DAE support: A major issue—that goes obviously far beyond the scope of buildings simulation—is the lack of proper support for DAE-based models in the FMI specification. To overcome this problem, a dedicated DAE interface specification has been proposed [22], whose feasibility has been demonstrated by successfully interfacing the VHDL-AMS modeling language. This work could serve as a reference implementation for future extensions of the FMI specification.

2) Building design and operation support: Research at G2ELab also focuses on co-simulation solutions that cover the full life cycle of buildings, i.e. different stages of the design and operation phases. According to this, a dedicated software component specification called *MUSE* (Model for Unified Energy Systems) [23] has been specified, which enables not only the simulation of buildings but also supports optimal sizing (at the design stage) and optimal operation (for embedded optimization). The main idea is to provide a building model that consists of several subcomponents—referred to as the *facets* of the model—which allow to reuse information at different stages according to the specific requirements, see Fig. 9. A *simulation facet* provides a dynamic model (e.g., a Modelica model or FMU) while a *sizing facet* offers a static calculation model and its Jacobian for coupling to an optimization algorithm [24]. A *management facet* contains the description of a linearized model, which can for instance be interpreted by dedicated optimizers [25]. Beside facets, *ports* are used to enable structural composition.

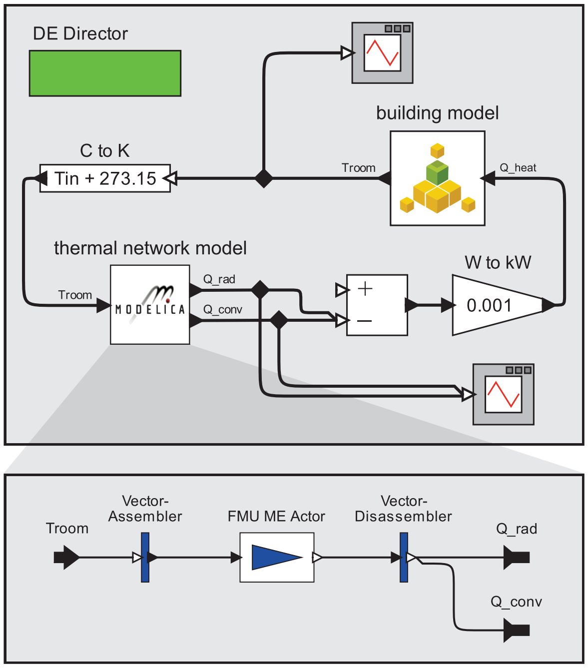

Fig. 8. Simple co-simulation setup. Figure taken from [20].

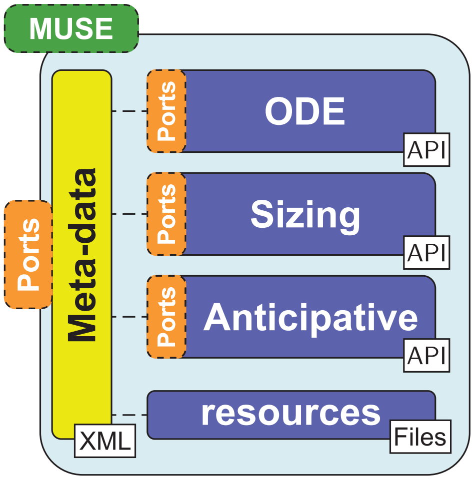

Fig. 9. Concept of the MUSE specification, including facets and ports.

Currently, the feasibility of the FMI specification for these purposes is examined. The first MUSE component prototypes that integrate FMUs have been developed, demonstrating the principle practicality. However, further studies will investigate this topic in more depth.

3) FMU deployment support: The MUSE specification has been designed with the objective to provide software components that can be easily deployed and reused. It is based on the OSGi specification[2], which facilitates the deployment of components and the management of their dependencies. In this regard the FMI specification could benefit from analogous approaches. Deploying FMUs would be facilitated by sharing wrappers that allow calling FMUs from languages different than C (e.g., Java, Python or C++). This may further support its dissemination and ease its integration into other software packages. An example of a software wrapper is the Java wrapper for FMI provided by JFMI[3] or the Python wrapper PyFMI[4]. An example for a software stack for deployment to the cloud is the FMQ project[5].

---

[2] See http://www.osgi.org/.

[3] See http://ptolemy.eecs.berkeley.edu/java/jfmi/.

[4] See http://www.pyfmi.org/.

[5] See http://www.xogeny.com/products/.

Regarding the simulation of buildings, an enhanced software component approach would facilitate the management of models for the full life cycle of a building and optimize model composition [26]. Indeed, cloud-based simulation dedicated to design and optimal management of buildings can be automatically deployed using MUSE software components [27].

III. CONCLUSION AND OUTLOOK

The presented methods focus to some extend on different key aspects, therefore using the concept of co-simulation and model exchange as a means to different ends. While some emphasize the vertical integration of domain-specific tools to allow more elaborate building simulations, others put priority on the development of more general simulation environments that enable a system-level view. However, regardless of these details, all approaches foster the integration of multiple domains and cyber-physical aspects, which become an ever more important part of buildings and community energy systems.

The presented applications also show clearly the benefits of including the FMI specification, since it enables the collaboration of different projects on a technical basis, promoting interoperability and reuse of existing implementations. However, a critical examination of these results also shows that efforts beyond the adherence to the FMI specification are necessary, driven by the actual needs of applied R&D projects. This includes especially work on extensions of the FMI specification, both in the form of proposing changes to the specification itself or by providing software solutions that work on top of it.

ACKNOWLEDGMENTS

The authors of this contribution are members of the IEA EBC Annex 60 collaboration.

This research was supported by the Austrian Research Promotion Agency (FFG), Project No. 843149.

This research was supported by the Assistant Secretary for Energy Efficiency and Renewable Energy, Office of Building Technologies of the US Department of Energy, under Contract No. DE-AC02-05CH11231.

We thank Pacific Controls for sponsoring the development of a Functional Mock-up Unit for co-simulation import interface in Niagara$^{\mathrm{AX}}$.

REFERENCES

[1] D. B. Crawley, J. W. Hand, M. Kummert, and B. T. Griffith, "Contrasting the Capabilities of Building Energy Performance Simulation Programs," *Buildings and Environment*, vol. 43, no. 4, pp. 661–673, 2008.

[2] P. Fritzson, *Introduction to Modeling and Simulation of Technical and Physical Systems with Modelica*, 1st ed. Wiley-IEEE Press, 2011.

[3] T. Blochwitz, M. Otter *et al.*, "The Functional Mockup Interface for Tool independent Exchange of Simulation Models," in *Proceedings of the 8th International Modelica Conference*, Dresden, Germany, 2011.

[4] S. A. Klein, J. A. Duffie, and W. A. Beckman, "TRNSYS: A transient simulation program," *ASHRAE Transactions*, vol. 82, pp. 623 – 633, 1976.

[5] V. Norrefeldt, G. Grün, and K. Sedlbauer, "VEPZO – Velocity propagating zonal model for the estimation of the airflow pattern and temperature distribution in a confined space," *Building and Environment*, vol. 48, pp. 183–194, 2012.

[6] V. Norrefeldt, "VEPZO – Velocity Propagating Zonal Model – A locally refined airflow model for confined spaces to use in optimization applications," Ph.D. dissertation, University of Stuttgart, 2013.

[7] S. Kübler, "Enhancement of Building Energy Simulation by Integrating Zonal Models for Airflow Computation in Atria," Master's thesis, TU München, 2013.

[8] W. Tian and W. Zuo, "Literature Review and Research Needs to Couple Building Energy and Airflow Simulation," in *Proceedings of the APEC Conference on Low-carbon Towns and Physical Energy Storage*, Changsha, China, 2011.

[9] M. Wetter, W. Zuo, T. S. Nouidui, and X. Pang, "Modelica Buildings Library," *Journal of Building Performance Simulation*, vol. 7, no. 4, 2014.

[10] W. Zuo and Q. Chen, "Real-time or Faster-than-real-time Simulation of Airflow in Buildings," *Indoor Air*, vol. 19, no. 1, pp. 33–44, 2009.

[11] ——, "Fast and Informative Flow Simulations in a Building by Using Fast Fluid Dynamics Model on Graphics Processing Unit," *Building and Environment*, vol. 45, no. 3, 2010.

[12] Tridium and Sun-Microsystems, *Baja: A Java-based Architecture Standard for the Building Automation Industry*, 2000. [Online]. Available: http://www.automatedbuildings.com/wsim/

[13] D. B. Crawley, L. K. Lawrie, F. C. Winkelmann, W. F. Buhl, Y. J. Huang, C. O. Pedersen, R. K. Strand, R. J. Liesen, D. E. Fisher, M. J. Witte, and J. Glazer, "EnergyPlus: creating a new-generation building energy simulation program," *Energy and Buildings*, vol. 33, no. 4, Apr. 2001.

[14] T. S. Nouidui, M. Wetter, and W. Zuo, "Functional mock-up unit for co-simulation import in EnergyPlus," *Journal of Building Performance Simulation*, vol. 7, no. 3, pp. 192–202, 2014.

[15] T. S. Nouidui, D. M. Lorenzetti, and M. Wetter, *EnergyPlusToFMU*, 2013. [Online]. Available: http://simulationresearch.lbl.gov/fmu/EnergyPlus/export/index.html

[16] M. Wetter, "Co-simulation of building energy and control systems with the Building Controls Virtual Test Bed," *Journal of Building Performance Simulation*, vol. 3, no. 4, pp. 185–203, 2011.

[17] T. S. Nouidui, M. Wetter, Z. Li, X. Pang, P. Bhattacharya, and P. Haves, "BACnet and Analog/Digital Interfaces of the Building Controls Virtual Test Bed," in *12th International Building Simulation Association Conference*, Sydney, Australia, 2011.

[18] E. A. Lee, S. Neuendorffer, and G. Zhou, *System Design, Modeling, and Simulation using Ptolemy II*. Ptolemy.org, 2014. [Online]. Available: http://ptolemy.org/books/Systems

[19] E. Widl, W. Müller, A. Elsheikh, M. Hörtenhuber, and P. Palensky, "The FMI++ library: A high-level utility package for FMI for Model Exchange," in *Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES)*, 2013, pp. 1–6.

[20] E. Widl, W. Müller, D. Basciotti, F. Stift, and P. Palensky, "Co-Simulation of Complex Energy Systems based on the Functional Mock-Up Interface Specification," *IEEE Transactions on Industrial Informatics*, 2014, submitted.

[21] E. Widl and W. Müller, "Linking FMI-based components with discrete event systems," in *Proceedings of the International IEEE Sytems Conference (SysCon)*, 2013.

[22] A. Rezgui, L. Gerbaud, and B. Delinchant, "Unified modeling technique using VHDL-AMS and software components," *Mathematics and Computers in Simulation*, pp. 266–276, 2013.

[23] S. Robert, B. Delinchant, B. Hilaire, and Y. Tanguy, "PLUMES: Towards a unified approach to building physical modeling," in *Proceedings of Building Simulation 2013*, Chambery, France, 2013.

[24] F. Verdière, A. Rezgui, S. Gaaloul, B. Delinchant, L. Gerbaud, F. Wurtz, and X. Brunotte, "Modelica models translation into Java components for optimization and DAE solving using automatic differentiation," in *Proceedings of IEEE UKSim2012*, Cambridge, England, 2012.

[25] G. Warkozek, D. Ploix, F. Wurtz, M. Jacomino, and B. Delinchant, "Problem formulation and analysis for optimal energy management in multisources systems: W effect," *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, pp. 904–919, 2012.

[26] S. Gaaloul, B. Delinchant, F. Wurtz, and F. Verdière, "Software Components For Dynamic Building Simulation," in *roceedings of Building Simulation 2011*, Sydney, Australia, 2011.

[27] B. Delinchant, P. Gibello, F. Verdière, and F. Wurtz, "Cloud computing services for the design and optimal management of buildings," in *Proceedings of Building Simulation 2013*, Chambery, France, 2013.